

Projektityön dokumentti – Älykäs reseptikirja

Yleiskuvaus

Älykäs reseptikirja on ohjelma, joka ylläpitää ruokareseptikirjaa, jonka avulla voi helposti etsiä kriteerit täyttävä ruokalaji. Käyttäjällä on ajanmukaiset tiedot ruoka-aineista, jotka ovat sillä hetkellä varastossa. Ohjelma voi etsiä ruokalajeja, jotka voidaan valmistaa käymättä kaupassa, tai ruokalajit, jotka vaativat N puuttuvaa tai osittain puuttuvaa ainetta. Reseptejä voidaan hakea myös niiden sisältämien ruoka-aineiden perusteella.

Aineisiin voidaan liittää merkintä niiden sisältämistä allergeeneistä, jotta voidaan rajoittaa hakua, jos halutaan välttää esimerkiksi maitoa sisältäviä ruokalajeja. Aineet voidaan myös rakentaa itse reseptistä samoin kuin ruokalajit, esimerkiksi jauhelihapihvin resepti voi sisältää lihamureketaikinaa, jolla on puolestaan oma reseptinsä. Jos lihamureketaikina puuttuu, ohjelma yrittää koota sen raaka-aineistaan. Ohjelma käsittelee sekä reseptejä että ruoka-aineita "aineina": reseptit ovat aineita, joita voidaan valmistaa muista aineista, joita kutsutaan ohjelmassa ainesosiksi. Ainesosat, joilla ei ole omia ainesosiaan, kutsutaan raaka-aineiksi.

Ohjelmassa on myös luokka, Muuntaja, joka tekee yksikkömuunnokset aineille niiden tiheyttä käyttäen. Ohjelmassa kaikkien aineiden tiheyden mittayksikkö on g/ml. Muuntaja on olennainen, koska resepteissä aineet voivat ilmetä eri mittayksikössä kuin tavallisesti. Esimerkiksi jauhoja voidaan ostaa kiloittain, mutta reseptit voivat mitata niitä desilitroina.

Ohjelmassa on merkkipohjainen käyttöliittymä, jolla voi hallita ohjelman varastoa ja luoda uusia aineita. Graafinen käyttöliittymä on ulkoasultaan valmis, mutta toiminnallisuus puuttuu. Kun ohjelmaa yrittää ajaa Eclipsessä, ilmenee "Error: Could not find or load main class GUI". Kysyin ongelmaan apua, mutta kirjoittamishetkellä en ole onnistunut ratkaisemaan ongelmaa esitetyillä korjausehdotuksilla. Ohjelma on siis pyritty toteuttamaan keskivaikealla tasolla, mutta olettaisin, että tällä hetkellä se tulee parhaimmillaan olemaan bumerangi.

Käyttöohje

Ohjelma käynnistetään GUI-objektin kautta. Siitä aukeaa ohjelman pääikkuna, jossa on taulukko, jossa on listattuna kaikki ohjelman tuntemat aineet ja niiden määrät. Ikkunassa on myös kolme nappia: Reseptihaku, Luo Resepti ja Varastohallinta. Näistä jokainen avaa ikkunan, joissa voidaan suorittaa erinäisiä toimintoja. Reseptihaussa voidaan hakea eri aineita, reseptinluonnissa voidaan luoda uusia aineita ja varaston hallinnassa voidaan muuttaa aineiden määriä ohjelman muistissa.

Aine-olioiden kaikkia muuttujia pystytään muokkaamaan eri metodeilla, käyttöliittymän Aine-näkymän kautta. Esimerkiksi ainesosia voidaan poistaa, lisätä ja muokata halutessa. Varaston tietoja eri aineista ja niiden määristä voidaan vastaavasti muuttaa Varastohallinta -ikkunasta. Aine- ja Varastotiedot voidaan myös lukea tekstitiedostoilta ja tallentaa niille.

Aineita voi etsiä Reseptihaku-ikkunan kautta. Siinä on tekstikenttiä, jolla voi asettaa hakukriteerejä. Haku palauttaa listan aineita, jotka täyttävät mahdollisesti annetut kriteerit.

Ohjelman rakenne

Ohjelma on jaettu kahdeksaan pääosaan: Aine-luokkaan, Hakukone-, Muuntaja-, Varasto-, IO- ja GUI-yksittäisolioihin sekä Poikkeukset poikkeusluokkiin. Näistä keskeisimmät ovat Aine, Varasto ja Hakukone; muut luokat lähinnä helpottavat näiden toimintaa.

Aine-luokka mallintaa reseptejä, ainesosia ja raaka-aineita, eli käytännössä kaikenlaisia aine-olioita. Aine-olioihin on tallennettu kaikki aineille ominainen tieto: nimi, allergeenit, tiheys, määrä (joka syntyy valmistettaessa), oletusmittayksikkö sekä ainesosat, eli mitä aineksia vaaditaan aineen valmistukseen. Lopuksi on myös aineen kuvaus, joka on yleisimmiten valmistusohje aineelle.

Varasto-olioon tallennetaan tiedot kaikista ohjelman tuntemista Aine-olioista, ja tieto kuinka paljon käyttäjällä on niitä varastossa. Varastolla on metodeja, joilla voidaan hallinoida näitä määriä ja välittää näitä tietoja eteenpäin erityisesti Hakukoneelle.

Hakukone-olio on työkalu, joka helpottaa reseptien löytämistä Varaston tiedoista. Sen metodeja käytetään palauttamaan ainelistoja. Aineita voidaan suodattaa nimen ja allergeenien perusteella. Kriteeriksi voidaan myös määritellä, että aine pitää pystyä valmistamaan varastossa olevista aineksista, tai siten että puuttuu n aineista.

Muuntaja-oliota käytetään tekemään muunnoksia eri massojen ja tilavuuksien mittayksiköiden välillä. Hakukone käyttää Muuntajaa tarkistaessaan riittävätkö Varaston ainekset, koska Varastossa olevat aineet voivat olla eri mittayksikössä kuin reseptissä jota Hakukone tarkistaa.

IO-olio käsittelee ohjelman tietojen tallennustarpeita. Se tallentaa Aine-olioita omille tekstitiedostoilleen ja Varaston tiedot yhdelle tekstitiedostolle. Se voi myös välittää tiedot Varastolle tältä tiedostolta, ja luoda Aine-tiedostoilta uusia Aine-olioita. IO suorittaa näitä toimenpiteitä, kun ohjelma käynnistetään tai suljetaan, tai kun luodaan uusia aineita käyttöliittymän kautta.

GUI-olio on ohjelman graafinen käyttöliittymä. Sen kautta voidaan käyttää kaikkia ohjelman toimintoja, ja huomattavasti kätevämmän kuin merkkipohjaisesti. Käyttöliittymässä on viisi suurempaa näkymää, muut osat ovat lähinnä ponnahdusikkunoita ynnä muita vastaavia. Pääikkunassa on listattuna taulukkoon kaikki Varaston tuntemat aineet ja niiden määrät, sekä napit joista pääsee kolmeen muuhun ikkunaan. Reseptin luonti-ikkunassa voidaan interaktiivisesti luoda uusia Aine-olioita, ja muuttaa olemassa olevia. Reseptihaussa voidaan käyttää Hakukonetta aineiden etsimiseen. Varastohallinnassa voidaan kätevästi muuttaa aineiden määriä Varastossa. Viimeinen ikkuna on Aineikkuna, jonka voi avata painamalla listattuja aineita eri näkymissä. Siinä voi tarkastella ja halutessa muuttaa aineen ominaisuuksia.

Algoritmit

Valtaosa Muuntaja-olion metodeista toimii käyttäen kaavaa $\text{tiheys} = \text{massa} / \text{tilavuus}$ ($d = m / V$). Tätä kaavaa hyödyntääkseen Muuntaja laskee eri mittayksiköiden suhteet (esim. $\text{kg/g} = 1000$), josta voi laskea muunnoksia, kun on kyse muunnos massayksiköstä massayksikköön tai vastaava tilavuuksien välillä. Tilavuuksien ja massojen välisiä muunnoksia varten yksiköt muutetaan vastaamaan Aine-olioiden käyttämää tiheyden yksikköä (g/ml).

Aine-luokka ja Hakukone-olio käyttävät rekursiota käydessään läpi Aine-olioiden ainesosia. Metodeita Aine.aineetYhteensä, Aine.sisältääAineen ja Hakukone.voiValmistaa kutsutaan siis ensin kyseessä olevalle Aine-oliolle, ja sitten sen mahdollisille ainesosille ja niin edelleen, kunnes saavutetaan aineen raaka-aineet.

Hyvin monet metodit käyttävät silmukoita käymään läpi kokoelmia.

Tietorakenteet

Varasto käyttää hakurakennetta (Map), koska se koostuu käytännössä täysin avain-arvo -pareista (aine-määrä). Muut tietorakenteet ovat lähinnä muuttuvatilaisia, koska halusin toteuttaa aineet siten, että niiden ominaisuuksia voi halutessa muuttaa tai korjata.

Tiedostot

Ohjelma käsittelee IO-olion kautta kahdenlaisia tekstitiedostoja: Aine- ja Varasto-tiedostoja. Aine-oliot tallennetaan tekstitiedostoihin, Reseptikirjan reseptikansioon. Sieltä ne voidaan lukea myöhemmin tarvittaessa. Tiedoston ensimmäiselle riville tulee aineen nimi. Toiselle riville tulee aineen tiheys, määrä ja mittayksikkö. Seuraaville riveille tulevat ainesosat, jokainen omalle rivilleen. Ainesosien jälkeen seuraavalla rivillä on tähtimerkki ("*") erottimena, jonka jälkeen seuraavalla rivillä on aineen allergeenit. Viimeisillä riveillä on aineen kuvaus. Esimerkki:

Spagetti bolognese

0.0,4.0,kpl

spagetti,300.0,g

kastike,800.0,g

liha,tomaatti

Spagetti bolognese, neljä annosta. Paista jauheliha... jne.

Varaston tiedot tallennetaan riveittäin muodossa nimi, määrä. Esimerkki:

spagetti,300.0,g

maito,1.0,l

kananmuna,12.0,kpl

Testaus

Ohjelman testaus oli hyvin puutteellista. Tämä johtui siitä, että en saanut scalatest-kirjastoa, jota olin suunnitellut käyttäväni yksikkötestaukseen, toimimaan projektissani kuin vasta toisen checkpointin jälkeen. Testaus oli hyvin pitkälti scala-tulkilla ja silmämääräisesti tarkistaen, että esimerkiksi tekstitiedostot luotiin oikein ja pystyttiin myös lukemaan. Ajan loputtua kesken, ehdin toteuttaa suunnittelemani yksikkötesteistä ainoastaan metodin aineetYhteensä toimintaa testaavia.

Ohjelmaan tuli enemmän metodeja kuin olin alun perin suunnitellut, joten alkuperäinen testaussuunnitelmani ei kattanut aivan koko ohjelmaa. Valitettavasti en ehtinyt toteuttaa edes kaikkea suunnittelemani testausta.

Ohjelman tunnetut puutteet ja viat

Ohjelma tämän dokumentin kirjoitushetkellä on puutteellinen. Suurin puute on toimiva käyttöliittymä, jota en ehtinyt toteuttaa, kun odottamattomat ongelmat kohtasivat huonon aikataulutukseni. Teen sen valmiiksi, mutta valitettavasti en ehdi ennen deadlinea, joten tämä palautus on puutteellinen.

Hakukoneen metodi voiValmistaa, jolla tarkistetaan voiko annetun aineen valmistaa varastossa olevilla aineilla, on puutteellinen. Tällä hetkellä se palauttaa oikean arvon, kun kyseessä on aine, jolla ei ole ainesosia, tai sen ainesosat ovat raaka-aineita (eli ainesosilla ei ole omia ainesosia). Väärä tulos saattaa tulla, kun esimerkiksi reseptiin A tarvitaan 2 ainetta B, jonka valmistamiseen puolestaan tarvitaan 4x C. Tämä virhe syntyy, koska metodi ei tee tarkistusta paljonko ainetta B syntyy kun sitä valmistetaan aineesta C. En keksinyt ratkaisua tähän ongelmaan.

Jos menee omakätisesti muuttamaan varasto-tekstitiedostoa, ja lisää sinne aineita, joita ei ole reseptikansiossa, Varaston tietoja ei pysty täyttämään kokonaan metodilla lataa(), koska syntyy poikkeus, kun yritetään lisätä ohjelmalle tuntematon aine.

Ohjelmassa ei ole ehkä tarpeeksi lisämäärittelyitä, kuten private, varsinkin koska suurinta osaa muuttujista voi joka tapauksessa muuttaa jollakin saman luokan metodilla.

En huomannut muita vikoja, mutta niitä on voinut päästä tarkistusteni ohi, koska en tehnyt kattavaa yksikkötestausta.

Kolme parasta ja kolme heikointa kohtaa

Mielestäni paras kohta ohjelmasta on tekstitiedostojen asettelu. Aine-tiedostoja pystyy helposti kirjoittamaan suoraan tekstitiedostolle, ja niitä on helppo lukea, kun tunnistaa muuttujien järjestyksen. Tämä oli hyödyllistä testatessa ohjelmaa ilman käyttöliittymää, sillä se oli huomattavasti nopeampaa kuin kirjoittaa konstruktoreita yhteen pötköön Aine-olioille.

Toinen vahva kohta on mielestäni koodin luettavuus. En ole kokenut koodari, joten arvioini voi olla hyvinkin pielessä, mutta kehitystä on tapahtunut verrattuna vanhempiin ohjelmiini. Mielestäni ohjelma on suurimmaksi osaksi selkeä lukea ja ymmärtää mitä eri metodeissa tapahtuu. En kommentoinut kaikkea täysin tyhjentävästi, mutta mielestäni nimesin metodit ja niiden konstruktorit selkeästi.

Kolmas vahva kohta on graafisen käyttöliittymän ulkoasu. En harmillisesti ehtinyt toteuttaa toiminnallisuudesta kuin vain pienen osan, mutta mielestäni siitä olisi tullut hyvin intuitiivinen ja interaktiivinen. Vahvuus mielestäni olisi ollut, että aineisiin olisi päässyt käsiksi kirjoittamatta mitään: tarkoitus oli, että missä ikinä taulukossa jokin aine oli, sitä painamalla pääsisi käsiksi aineen tietoihin. Aineikkunasta puolestaan olisi kätevä muuttaa eri ominaisuuksia halutessaan.

Valitettavasti käyttöliittymä on myös ohjelman suurin heikkous. Sen toiminnallisuus jäi puolitiehen. Ohjelmaa ei käytännössä pysty käyttämään niin kuin oli suunniteltu.

Toinen heikkous on Hakukone. Se täyttää suurimman osan toiminnallisuudesta, mutta niin kuin mainittu viat ja puutteet -kohdassa, siinä on tapauksia, jossa se ei toimi täysin oikein. Puutteet eivät välttämättä ilmaannu arkisemmissa ruokalajeissa, mutta jos lähdetään lisäämään moniasteisia reseptejä, virheellisten hakutulosten riski kasvaa.

Kolmas heikkous on Aine-olioiden ja Varaston toiminta tekstitiedostojen kanssa. Tekstitiedostoja on helppo korruptoida, ja kuten aiemmin mainittu tämä hankaloittaa eritoten Varaston luomista. Aine-oliot ovat itsenäisiä, mutta jos yritetään löytää Varastolle olemattomia aineita, toiminta hankaloituu.

Poikkeamat suunnitelmasta ja aikataulu

Alun perin suunnittelin, että ainesosat-muuttuja on Aine-luokan konstruktori, mutta siirsin sen tavalliseksi muuttujaksi. Tämä oli sen vuoksi, että Aine-olioiden luominen on huomattavasti hankalampaa, jos ohjelman täytyy tietää aineen kaikki ainesosat ennen luomista. On helpompaa luoda aine ensin, ja luoda ainesosat halutessa, ja sitten lisätä ne reseptiin.

Päädyin assistentin neuvosta myös jakamaan IO-yksittäisolion erilleen, mikä oli hyvä idea.

Ajankäyttösuunnitelmani ei osunut ollenkaan kohdilleen. Minulla meni Aine-, Varasto- ja Hakukone-luokissa luultavasti noin puolet siitä ajasta, jonka arvioin. Sen sijaan Muuntaja-luokassa kesti muutama tunti pitempään kuin suunnittelin. Graafisessa käyttöliittymässä luulisin, että olisin ollut noin arviossani, mutta en osannut odottaa että joutuisin painimaan Eclipsen kanssa 5-10 tuntia. Minulla tuli eteen ongelmia, jotka pysäyttivät täysin työskentelyni etenkin käyttöliittymän kanssa. Sain toteutettua muut luokat sen verran hyvässä tahdissa, etten varannut tarpeeksi aikaa käyttöliittymää varten.

Lähdin työstämään projektia liian myöhään, kuten sain huomata viimeistään tätä dokumenttia kirjoittaessani. Työjärjestykseni oli suurin piirtein sama kuin suunnittelin, poikkeuksena Muuntaja, jonka toteutin ennen kuin viimeistelin Aine-luokan.

Arvio lopputuloksesta

Ohjelma on selvästi puutteellinen tällä hetkellä käyttöliittymän keskeneräisyyden vuoksi, mutta koska palautusaika tuli eteen, joudun palauttamaan sen tällaisena. Aion tehdä ohjelman loppuun ja palauttaa sen kokonaisena, palautusjärjestelmän salliessa.

Mielestäni ohjelmani olisi täysin tyydyttävä (jos sitä voisi käyttää). Mielestäni lukuunottamatta Hakukoneen lieviä puutteita ja Varaston korruptioherkkyyttä, ohjelma täyttää toiminnallisuusvaatimukset. Uskon, että käyttöliittymä on hyvin kätevä, kunhan viimeistelen sen. Uskoakseni mallinnukseni ja tietorakennevalinnat sopivat ohjelman toteutukseen. Toteutin kattavasti metodeja kaikille luokille, joten erityisesti käyttöliittymää lienee helppo laajentaa suunnittelemastani, halutessaan. Jos olisin taitavampi ruoanlaittaja, käyttäisin ohjelmaa arjessa mielelläni.

Viitteet

Ohjelmointi 1 -kurssimateriaali
Ohjelmointistudio 2 -kurssimateriaali
Ohjelmointi 2 -kurssimateriaali
scala API
scala.swing API
stackexchange.com
tutorialspoint.com/scala

Liitteet

Smart_Cookbook -Scala-projekti